VISIÓN INVESTIGADORA

ROBOT MÓVIL CONTROLADO POR COMANDOS DE VOZ LPC-DTW

MOBILE ROBOT CONTROLLED BY MEANS OF LPC – DTW VOICE COMMADS

Yeison H. Baquero[1]
Zuleika Alezones[2]
Henry Borrero[3]

Fecha de envío: Noviembre de 2010
Fecha de recepción: Diciembre de 2010
Fecha de aceptación: Marzo de 2011

Resumen:

La utilización de herramientas computacionales, hardware y software con capacidades de desarrollo en torno al manejo de modelos bioinspirados, generó la construcción de un minirobot móvil aplicando la caracterización de comandos de voz y el aprovechamiento de las capacidades de procesamiento paralelo ofrecidas por el gene digital, para gobernar la correspondiente navegación. A nivel general, en la etapa de software se implementó un prototipo de programa en el cual se presenta el reconocimiento de palabras aisladas en castellano emitidas por un locutor, aplicado al control de navegación de un minirobot. La alternativa cuenta con un desarrollo adelantado con el lenguaje Java y consta de cuatro módulos: obtención de la señal hablada, extracción de características, comparación de características, procesamiento de los comandos caracterizados por medio de un gene digital y comunicación de las acciones de control a los actuadores del minirobot.

Palabras clave:

LPC, DTW, robot móvil, descriptores.

Abstract:

The use of computational tools, hardware and software capable of handling and developing bio-inspired models led to the construction of a mobile mini-robot using voice-commands characterization as well as the parallel processing capabilities, offered by the digital gene, in order to govern the corresponding navigation. In the software development stage, a program prototype that shows the isolated speaker-produced word recognition in Spanish was implemented to lead the cruise control of a mini-robot. The application involved advanced development using java language and consists of four modules, namely acquisition of the speech signal, feature extraction, feature comparison, and processing of commands characterized using a digital gene and also using control-actions communication with the mini-robot actuators.

Keywords:

LPC, DTW, mobile robot, descriptors.

1 Ingeniera de Sistemas, Universidad de los Llanos. Auxiliar de investigación: Grupo de Investigación en Robótica (GIRO), Universidad de los Llanos. Correo: yeison@ingenieros.com

2 Ingeniero de Sistemas, Universidad de los Llanos. Grupo de Investigación en Robótica (GIRO), Universidad de los Llanos. Correo: zuleika@ingenieros.com

3 Ingeniero Electrónico. Especialista en Automática e Informática Industrial. Docente Universidad de los Llanos. Director Grupo de Investigación en Robótica (GIRO). Correo: h_borrelo@ieee.org

1. Introducción

En el desarrollo de un software reconocedor de voz se deben tener en cuenta aspectos como: frecuencia de muestro de la señal de voz según criterio de Nyquist [1]; método de extracción de las características que describen la señal de voz (descriptores); método de comparación de patrones. Básicamente se implementó un prototipo de programa que al utilizarlo permite reconocer palabras aisladas en lengua castellana por parte de un locutor, aplicando esto al control de navegación de un minirobot móvil. La aplicación cuenta con un desarrollo adelantado en el lenguaje Java y consta de los siguientes cinco módulos: obtención de la señal hablada, extracción de características utilizando el método de codificación por predicción lineal (LPC, *linear predictive coding*), comparación de características con el método de alineación temporal dinámica (DTW, *dinamyc time warping*), procesamiento de los comandos caracterizados por medio de un gene digital y comunicación de las acciones de control a los actuadores del minirobot.

El documento que se presenta expone los avances logrados en el marco de un proyecto de investigación oficialmente en ejecución por parte del Grupo de Investigación en Robótica (GIRO) de la Universidad de los Llanos. Se presentan los aspectos generales teóricos pertinentes a cada una de las etapas en el reconocimiento computacional de comandos de voz y el gene digital, así las generalidades de la implementación y puesta en marcha.

En el contexto del reconocimiento de comandos de voz es necesario captar la señal que corresponde y realizar una serie de etapas para adecuar la señal.

2. Preprocesamiento

Una señal de voz presenta por naturaleza una atenuación en las frecuencias altas, por lo cual se debe realizar un filtraje que permita obtener información suficiente de esas frecuencias para no concentrarse únicamente en la información vinculada a las frecuencias bajas. Hay que tener en cuenta que el oído humano es más sensible a frecuencias en la zona de los 3.000 Hertz. La aplicación de un filtro preénfasis, representado por la ecuación 1, ayuda a que el procesamiento de la señal sea menos susceptible a truncamientos.

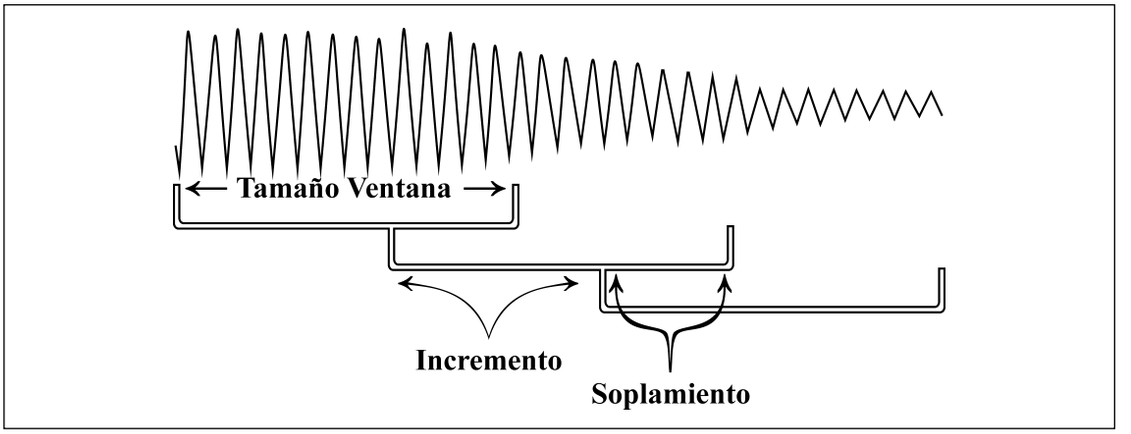
$$y[n] = x[n] - ([n-1]) \qquad (1)$$

En la ecuación 1 se tiene que $x[n]$ es el vector de amplitud de la señal de voz de entrada, $y[n]$ es la señal de salida del filtro preénfasis; entonces, si $\alpha < 0$, tenemos un filtro de paso bajo, y si $\alpha > 0$, un filtro de paso alto; para nuestro fin se utiliza $\alpha = 0{,}97$.

3. Extracción de características (descriptores)

Teniendo claro que la unidad básica del habla es la que se pretende reconocer (fonemas, vocales, silabas, palabras, frases, etc.) [1], se trabaja con un conjunto de grabaciones realizadas en un ambiente adecuado (muestras), las cuales se preprocesan, para proceder a aplicar algún método de extracción de características sobre la señal de voz. Debido a que las ondas de voz contienen numerosas variaciones (suma de distintas frecuencias), el proceso de extracción de características se realiza a intervalos cortos de tiempo. Comúnmente los estudios de voz se realizan a intervalos de 20 y 30 milisegundos, que es cuando se considera que la onda de voz no presenta demasiados cambios. Este proceso

Figura 1. Ventaneo de una señal



se conoce como ventaneo de la señal [1] y se ilustra en la Figura 1.

Para el caso expuesto, se utiliza una frecuencia de muestreo de 22.050 Hz, se realiza un análisis a intervalos de 30 ms (lo que equivale aproximadamente a 661 muestras por ventana), con un incremento del 50%.

Una vez elegido el tamaño de ventana, a cada una se le asigna una función, con el fin de disminuir la importancia de los valores que se encuentran en los extremos de las ventanas, y evitar que características de estos valores varíen la interpretación de la parte central del bloque, que es la más significativa. Para tal fin, los tipos de funciones ventana más utilizados son el tipo Hann y el tipo Hamming [1].

Después de realizado el correspondiente ventaneo, los segmentos de voz son aptos para aplicarles técnicas de extracción de patrones, como es el caso de la codificación por predicción lineal (LPC, linear predictive coding), basada en la producción del habla. Se utiliza esta debido a que: proporciona un modelo adecuado de la señal de voz, sus parámetros se ajustan a las características del tracto vocal, representa la envolvente espectral de la señal de forma comprimida, los parámetros obtenidos mediante predicción lineal muestran un espectro suavizado que proporciona la información más representativa de la voz y es un método preciso y adecuado para computación, tanto por su sencillez como por su rapidez de ejecución.

El concepto básico de predicción lineal LPC se centra en que una muestra de una señal de voz $x(n)$ puede ser predicha por las k muestras anteriores de la misma señal, generando una señal aproximada $\tilde{x}(n)$, representada por medio de la ecuación 2.

$$\tilde{x}(n) = \sum_{i=1}^{k} \alpha_i * x(n - i) \qquad (2)$$

Se tiene como definición del error de predicción lo representado en la ecuación 5. Para hallar los coeficientes α_i de la ecuación (3) minimizando el error, se aplican mínimos cuadrados al intervalo de N muestras que se desee considerar.

$$e(n) = x(n) - \tilde{x}(n) \qquad (3)$$

En el contexto del reconocimiento de comandos de voz es necesario captar la señal que corresponde y realizar una serie de etapas para adecuarla.

4. Reconocimiento

Para la etapa de reconocimiento se tiene establecido el vocabulario del sistema (adelante, atrás, izquierda, derecha) y con él los parámetros LPC. La fase de reconocimiento se inicia con la palabra pronunciada por el locutor, la cual se parametriza del mismo modo; después el correspondiente patrón LPC se compara con los patrones de referencia previamente almacenados en memoria, usando una medida de similitud (Hamming, euclidiana, distancia máxima); la medida de distancia entre los parámetros usados es la euclidiana (4) estudiada en [2]:

Parámetros $P=(p_1,p_2,\ldots p_n)$ y $Q=(q_1,q_2,\ldots q_n)$

$$d=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\cdots+(p_n-q_n)^2} \qquad (4)$$

$$d=\sqrt{\sum_{i=1}^{k}(p_i-q_i)^2}$$

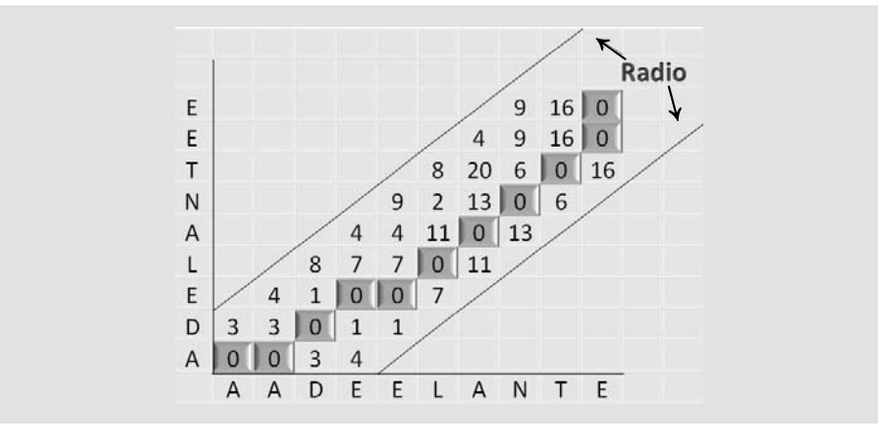
Debido a la variabilidad intralocutor de la señal, hay diferencias no lineales en la duración de los sonidos y la velocidad de pronunciación de los mismos, incluso tratándose de la misma palabra. Por tanto, se realiza un alineamiento temporal de los patrones (los recién calculados y los almacenados en memoria correspondientes a cada palabra del vocabulario), el cual consiste en minimizar la distancia total entre los estos.

Para realizar el alineamiento se utiliza el método de DTW (*dynamic time warping*) estudiado en [3, 4], el cual parte de determinar el patrón más similar a la palabra pronunciada, es decir, el que proporciona una menor distancia (distancia euclidiana) en la etapa de comparación. De manera más explícita, la comparación se realiza con cada palabra del vocabulario, generando un plano, uno de cuyos ejes se conforma con los parámetros calculados y el otro con los parámetros almacenados, donde cada punto o intersección en el plano es la distancia euclidiana calculada. Teniendo como finalidad encontrar la ruta mínima D desde el origen hasta la última intersección de ambos ejes, mediante la suma de las distancias de la diagonal en el plano (ecuación 5), tenemos:

$$D=\sum d \qquad (5)$$

Para lograr mejores resultados y evitar que la ruta siga en forma vertical o diagonal, se

Figura 2. Representación espacial en el plano generado por las características de la palabra adelante.



realiza un límite diagonal o radio, de manera que los valores seleccionados en el cálculo de las distancias e implementación del algoritmo están más cerca de la diagonal, como se muestra en la Figura 2.

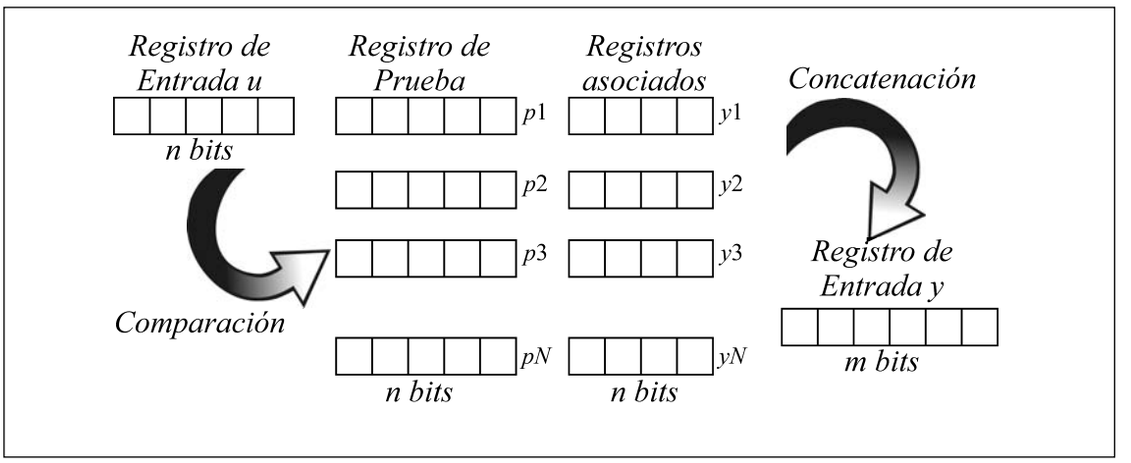
5. ADN y Gene digital

El gene digital consiste fundamentalmente en un acercamiento al gene biológico mediante emulación electrónica, para el correspondiente aprovechamiento en aplicaciones que requieran procesamiento paralelo. Los fundamentos presentados en el presente documento se basan en los adelantos conseguidos con el chip ADN emulado electrónicamente y el gene digital [5, 6, 7].

Tomando como referencia la Figura 3, el gene digital está compuesto por cuatro secciones: 1. un registro de entrada (*u*); 2. una serie de registros de prueba (*pi*); 3. registros asociados a los anteriores (*yi*); 4. un registro de salida (*y*). La operación consiste en realizar comparaciones simultáneamente entre el registro de entrada y cada uno de los registros de prueba. A partir de esta comparación, y según alguna restricción, el registro asociado correspondiente se concatena o no al registro de salida. Como método para realizar la comparación se propone el uso de la distancia de Hamming, en la que se mide el número de bits diferentes entre los registros. Para determinar la condición de concatenación, se define un parámetro llamado umbral de Hamming. Si la distancia es menor al umbral, se concatena el registro asociado correspondiente.

En la Figura 4 se muestra un diagrama general sobre la arquitectura en la que se basa el desarrollo del programa prototipo; básicamente, se emula el paralelismo disponible en el gene digital. Como se aprecia, la palabra de entrada (muestra) –que corresponde a una cadena de bits asociada a un comando de voz caracterizado como vector binario– se compara de manera paralela con un cierto número de palabras cargadas en los registros P_{r1} a P_{rN}. Las palabras almacenadas en los registros P_{r1} a P_{rN} corresponden al complemento de cada una

Figura 3. Diagrama de registros en el gene digital.



El gene digital consiste fundamentalmente en un acercamiento al gene biológico mediante emulación electrónica, para el correspondiente aprovechamiento en aplicaciones que requieran procesamiento paralelo.

Figura 4. Diagrama gene digital.

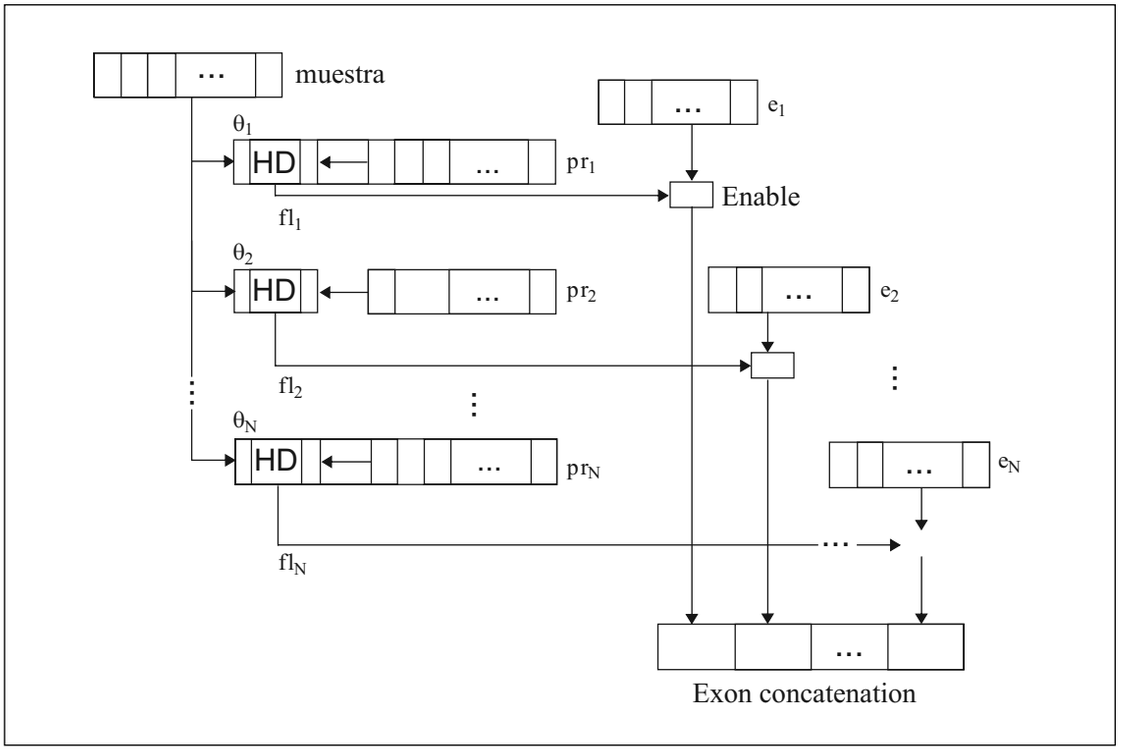
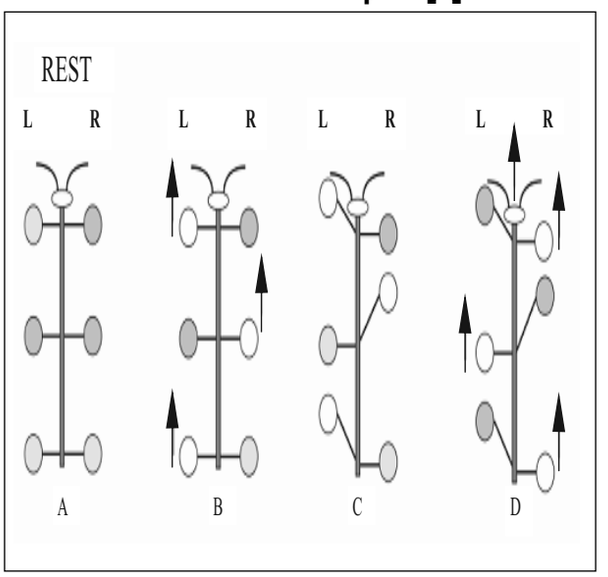


Figura 5. Representación sobre el modelo trípode para la locomoción del hexápodo [6].



de las palabras caracterizadas como comandos de voz. Como se aprecia, la muestra se procesa paralelamente con las palabras almacenadas en los registros, y se debe detectar la distancia Hamming entre cada una de estas palabras y la muestra. Si alguna de las palabras almacenadas sobrepasa un umbral θ_N, habilita el bloque *Enable* que le corresponde y, por tanto, la palabra e_N (acción de control) pasa al exón.

Las palabras cargadas en cada uno de los registros e_N corresponden a las acciones de control para navegación del minirobot.

6. Cinemática del robot

La locomoción del minirobot se basa en un modelo trípode de movimiento que, igual como sucede con los seres vivos, debe ser capaz de soportar su propio peso y superar la fuerza de gravedad. Este modelo básicamente persigue mantener tres patas en el suelo y darles libertad de movimiento a las demás. Una ventaja de este modelo de patas, es la estabilidad que se consigue para el minirobot y que permite aislar el cuerpo del terreno empleando puntos discretos de soporte. Así mismo, mediante patas es posible conseguir cierta omnidireccionalidad, con deslizamiento en la locomoción mucho menor [6].

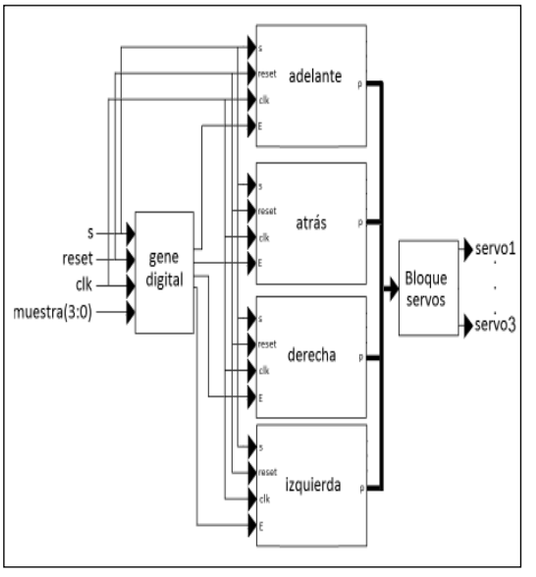
Como se muestra en la Figura 5A, la posición inicial del prototipo mantiene todas sus patas en el suelo; enseguida (Figura 5B) se reafirma la posición fija para tres de las patas, y las demás avanzan; en el siguiente paso se fijan las patas que avanzaron, lo que permite el avance de las demás. Esta dinámica de movimiento será reiterativa hasta el momento en que se reinicia el algoritmo que controla la cinemática del robot. En la Figura 6 se muestra una imagen de la estructura mecánica implementada para el funcionamiento del robot móvil.

7. Implementación

La aplicación del prototipo se encuentra desarrollada en lenguaje Java y se entrenó con las palabras “adelante”, “atrás”, “derecha” e “izquierda” de un mismo locutor. Cada palabra reconocida se caracterizó como vector binario que corresponde a la palabra de entrada (muestra) al gene-digital. En cuatro registros (P_{RN}) del gene-digital se encuentran almacenadas las palabras (vectores binarios) que generan un umbral adecuado para activar las acciones de control residentes en los registros e_n, como se puede apreciar en la Figura 7.

Figura 6. Prototipo minirobot.

Figura 7. Diagrama de bloques relacionado con desplazamientos del minirobot.



La locomoción del minirobot se basa en un modelo trípode de movimiento que, igual como sucede con los seres vivos, debe ser capaz de soportar su propio peso y superar la fuerza de gravedad.

Las palabras de control que residen en el exón de salida del gene digital se deben trasmitir a las respectivas entradas de los actuadores del minirobot (motores). Es importante resaltar que las palabras de control residentes en el exón de salida corresponden con las acciones de desplazamiento del prototipo. De acuerdo con el diagrama de la Figura 7, las salidas del gene digital corresponden al exón de salida del gene digital, de manera que se habilita cada uno de los bloques que habilita alguna de las dinámicas de desplazamiento de la figura; así mismo, cada uno de los bloques (adelante, atrás, derecha, izquierda) se habilita desde el gene digital, y estos habilitan el funcionamiento de los actuadores de salida: en principio, son tres servomotores o servos, como actuadores de un robot hexápodo (tres grados de libertad).

Retomando el diagrama de la Figura 7, se puede apreciar que los bloques correspondientes a la ejecución de los desplazamientos del minirobot corresponden en sí a una entidad que tiene como puertos de entrada: el puerto *s*, que permite poner en funcionamiento la dinámica del robot; el puerto *reset*, que simplemente permite reiniciar la dinámica del minirobot de manera asincrónica; y el puerto de entrada *muestra*, el cual corresponde a las palabras caracterizadas en el bloque de reconocimiento de comandos de voz; se cuenta también con tres (3) puertos de salida por medio de los cuales se transmiten señales de onda cuadrada con ciclo útil controlado, necesarias para el posicionamiento de los ejes de tres servos que hacen parte fundamental de los tres grados de libertad del sistema.

En cuanto a la aplicación software, la interfaz gráfica consta de tres componentes: el primero, destinado a la captura o carga de audio; el segundo, para extracción de características y reconocimiento de voz, y el último, de análisis. Adicionalmente, cuenta con un panel inferior en el cual se visualizan los resultados de aquellos procesos que lo requieran (Figura 8), mostrando a nivel interno (Figura 9) las etapas en el reconocimiento de voz, con los tipos de datos que se manejan en forma específica para la aplicación:

En la primera parte se obtiene el vector de voz en el tiempo, ya sea desde el módulo del

Figura 8. Interfaz gráfica, aplicación de reconocimiento de voz.

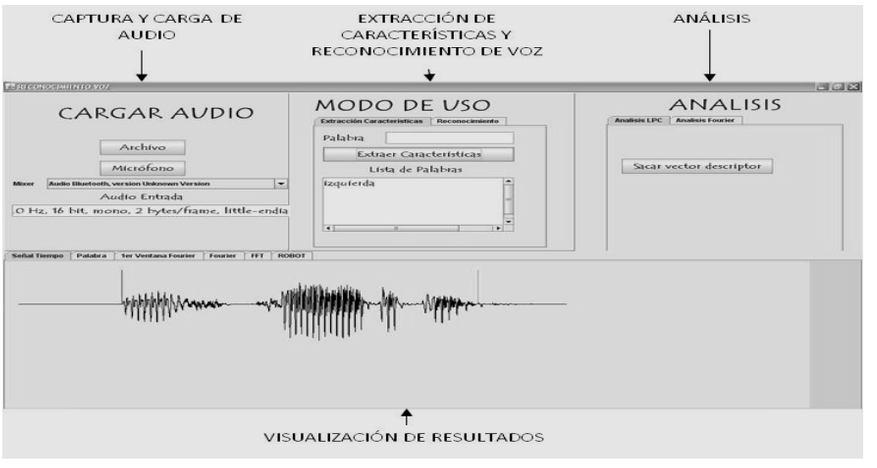
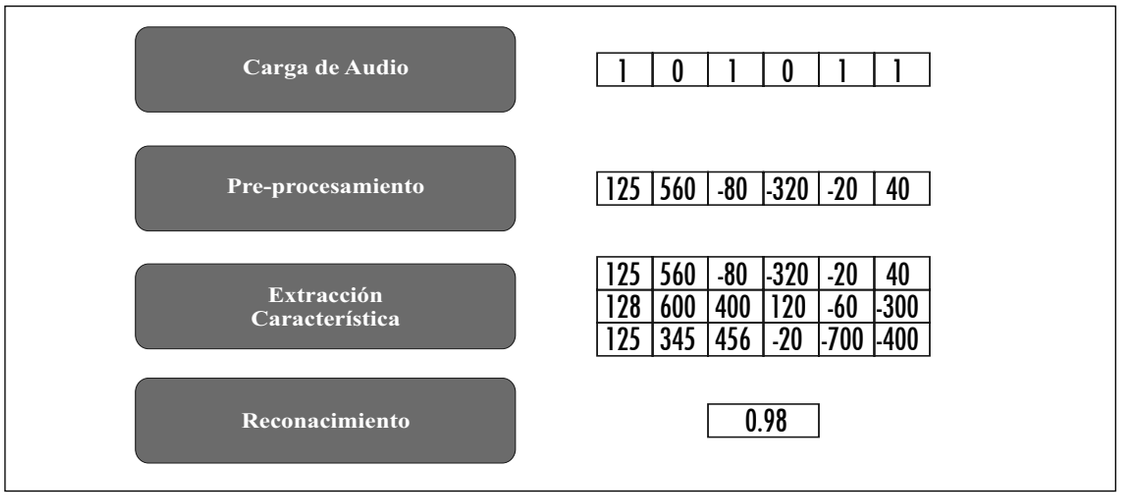


Figura 9. Salida de la información en cada fase del sistema.



micrófono o desde archivo. Esta información se entrega de forma binaria, con lo cual hay que realizar el respectivo *cast* para poder trabajarlo directamente en las posteriores etapas. En la siguiente etapa realizamos el filtro preénfasis sobre los datos de voz. Posteriormente, se obtienen los descriptores por cada ventana de voz, que en conjunto nos darán una matriz de elementos descriptores de la palabra, y dependiendo de si el siguiente paso es de reconocimiento, las características se guardan o no, para ser comparadas.

Por último, se comparan las características de las palabras guardadas y de las que se desea comparar, lo que mostrará cuál arroja la menor distancia y, por consiguiente, cuál será la palabra reconocida.

8. Resultados

La aplicación se probó en una población de 25 hombres y 25 mujeres (6 menores de 12 años, 11 de 12 a 19 años, 12 de 20 a 35 años y 21 de mayores de 35 años), de los cuales se tomaron cinco (5) muestras de cada palabra ("adelante", "atrás", "derecha" e "izquierda", adicionando además el reconocimiento de la palabra "alto" para un desarrollo posterior de una asociación para respuesta). En total, 25 muestras por persona (adelante 1-2-3-4-5, atrás 1-2-3-4-5, izquierda 1-2-3-4-5, derecha 1-2-3-4-5 y alto 1-2-3-4-5). Probando inicialmente el sistema mediante el entrenamiento de las muestras 1, y recopilando los resultados mediante el reconocimiento de las cuatro restantes, posteriormente se realiza la misma acción de entrenamiento para las muestras 2, 3, 4, 5 y sus respectivas pruebas con las muestras sobrantes.

Obteniendo de los análisis resultados de reconocimiento de 86,8% para los hombres (Figura 10), 82% para las mujeres (Figura 11) y un total de 84,45% de reconocimiento teniendo en cuenta la población en general (Figura 12). Las muestras se tomaron en un ambiente bastante natural, por tanto, la efectividad de la aplicación, que reside en un 84,45%, puede aumentar considerablemente, si se tienen en cuenta condiciones lo más ideales posible.

Las muestras fueron tomadas de manera continua, es decir, en grabaciones donde

En cuanto a la aplicación software, la interfaz gráfica consta de tres componentes: el primero, destinado a la captura o carga de audio; el segundo, para extracción de características y reconocimiento de voz, y el último, de análisis.

Figura 10. Porcentaje de reconocimiento para hombres.

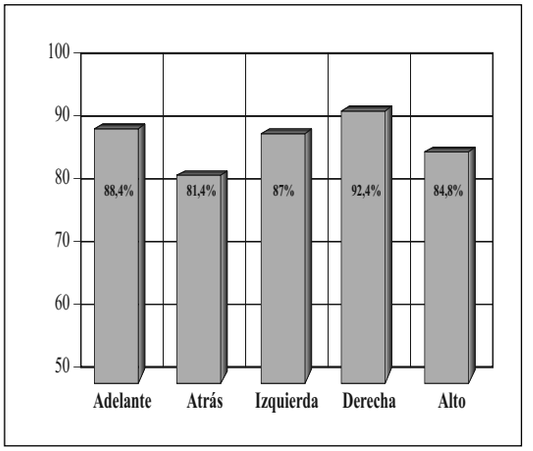
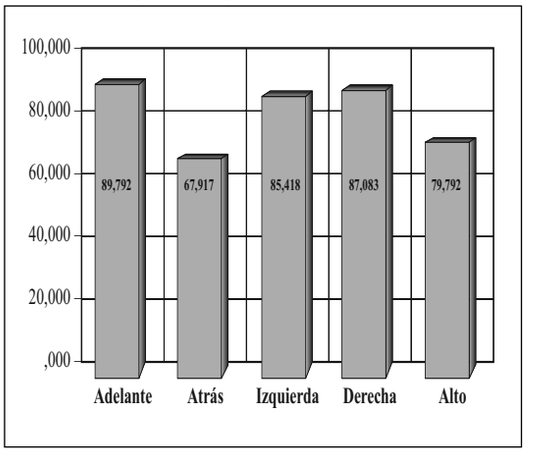
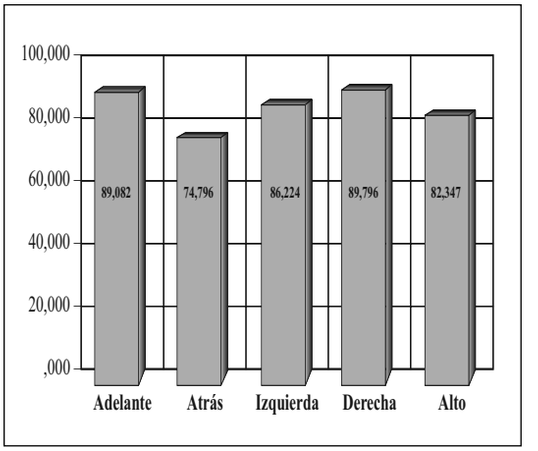


Figura 11. Porcentaje de reconocimiento para mujeres.



el locutor repetía una palabra varias veces, y posterior a ello se recortaron las palabras para formar conjuntos de muestras, lo que generó posibles errores humanos al momento de manipularlas. A pesar de las condiciones adversas en que fueron tomadas y manipuladas las muestras, se nota un porcentaje de reconocimiento bastante aceptable para el sistema, teniendo en cuenta, además, la poca experiencia de los desarrolladores en el tema.

Figura 12. Porcentaje de reconocimiento total de la aplicación.



Al momento de desarrollar un reconocedor de voz sin filtros de ruido, hay que tener en cuenta el ambiente al cual se va a aplicar, ya que en este caso el ruido sería necesario, porque al reconocer las muestras, estas presentarán un mayor grado de semejanza.

9. Conclusiones

Es necesario analizar las ondas de manera segmentada para comprender su evolución en el tiempo, ya que, si se usan tamaños de ventana demasiado grandes, se omiten cambios locales, contrario a si se toman tamaños de ventanas demasiado pequeños, ya que se reflejan demasiado los cambios puntuales.

El tamaño del incremento entre ventanas influye directamente en los tiempos de repuesta de los algoritmos y, a su vez, en la calidad de los resultados; con un incremento demasiado pequeño, el tiempo de respuesta es mayor y los resultados poco favorables.

El método LPC es adecuado al tratamiento del habla, ya que se aproxima a la producción de la misma.

El uso del algoritmo de programación dinámica (DTW) es ideal para el reconocimiento de señales de voz, porque trata de reducir las diferencias temporales naturales del habla.

El algoritmo DTW ofrece buenos resultados para un conjunto pequeño de palabras a reconocer; si se requiere realizar el reconocimiento para un vocabulario extenso, esta solución no es la más óptima computacionalmente.

El entrenamiento con características de uno o pocos hablantes hace que el reconocimiento de voz dependa del hablante, y para un reconocedor de voz general se deben realizar estudios con una muestra considerable de distintas voces, tratando de analizar características generales.

La primera etapa de realización del reconocimiento de comandos de voz y gene digital se efectúa en Java, como prototipo, para posteriormente implementarlo en hardware.

Vale la pena explotar las capacidades del chip ADN emulado electrónicamente, así como el gene digital y los algoritmos genéticos en la caracterización y reconocimiento de comandos de voz en un sistema embebido, aprovechando la disponibilidad de dispositivos lógicos reconfigurables, como las FPGA, pues cuentan con la posibilidad de procesar la información de comando de manera paralela y asociativa.

Es recomendable de adelantar trabajos de aplicación de filtros para el tratamiento adecuado del ruido en preprocesamiento, implementación de algoritmos diferentes en los diversos procesos de extracción de características y reconocimiento, sin ser dependiente en gran escala del hablante; entre muchos más trabajos futuros.

Referencias

[1] J. Bernal, S. Bobadilla, P. Gómez. "Reconocimiento de voz y fonética acústica". Madrid: RA-MA, 2000.

[2] A. Bregón y A. Alfonso. "Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos". Universidad de Valladolid, 2005.

[3] Eamonn J. Keogh, Michael J. Pazzani. "Derivative Dynamic Time Warping". 2001. En línea: http://www.cs.rutgers.edu/~mlittman/courses/lightai03/DDTW-2001.pdf

[4] J. Alvarado. "Reconocimiento de palabras aisladas utilizando MFCC y Dinamic Time Warping". Universidad Nacional de Trujillo, 2008.

[5] A. Malcolm Campbell y Laurie J. Heyer. Discovering genomics, proteomics, and bioinformatics. 2 ed. Cold Spring Harbor Laboratory Press y Benjamin Cummings, 2006, 447 pp.

[6] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter. Essential cell biology. Nueva York: Garland Science, 1998.

[7] J. Prieto, O. Ramos y A. Delgado. "Diseño de un gene digital en FPGA y MatLab con aplicaciones en robótica móvil". XIII Taller Iberchip IWS-2007, Lima, 14 a 16 de marzo de 2007.

[8] A. Farfán, J. Herreño y A. Delgado. "Gene digital y chip ADN electrónico: aplicaciones en robótica móvil". 3rd Colombian Workshop on Robotics and Automation (CWRA), Universidad Tecnológica de Bolívar, Cartagena, 21 a 22 de agosto de 2007.

Las muestras fueron tomadas de manera continua, es decir, en grabaciones donde el locutor repetía una palabra varias veces, y posterior a ello se recortaron las palabras para formar conjuntos de muestras, lo que generó posibles errores humanos al momento de manipularlas.